UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/781,867	02/20/2004	Edward Colles Nevill	550-513	5181

23117 7590 09/04/2007

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
MCFADDEN, MICHAEL B	
ART UNIT	**PAPER NUMBER**
2188	

MAIL DATE	DELIVERY MODE
09/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/781,867
Filing Date: February 20, 2004
Appellant(s): NEVILL, EDWARD COLLES

Stanley C. Spooner
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12 April 2007 appealing from the Office action mailed 20 October 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Wilson, Paul R. "Uniprocessor garbage collection techniques" in Proc. of International Workshop on Memory Management in Lecture Notes on Computer

Science. Springer-Verlag. Volume 637. (September 1992) pp 1-67. http://citeseer.ist.psu.edu/wilson92uniprocessor.html.

Hosoya et al. "Garbage Collection via Dynamic Type Inference - A Formal Treatment -" in Types in Compilation: Second International Workshop in Lecture Notes in Computer Science. Springer Berlin / Heidelberg. Volume 1473. (March 1998) pp. 215-239.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-6, 9, 11-16, 19, 21-26, and 29 are rejected under 35 U.S.C. 102(b) as being anticipated by Wilson ("Uniprocessor Garbage Collection Techniques").

3. **Regarding Claim 1, 11, and 21,** Wilson discloses a method of controlling execution of a processing task within a processing system, said method comprising the steps of: executing said processing task including allocating memory areas for data storage, and suspending an actual execution path of said processing task at an execution point to perform memory management said memory management comprising the steps of: identifying at least one data item roots occurring in the course of execution

and accessible to said processing task at said execution point which specify reference values pointing to respective ones of said memory areas **(graph of pointer relationships)**; determining a correlation between reference values corresponding to said at least one data item roots and memory areas allocated during said execution up to said execution point **(graph of pointer relationships)** by identifying at least one data item reachable from said at least one data item roots; and performing a memory management operation on allocated memory areas in dependence upon said correlation. **(all citations from Wilson: Page 9, Section 2.2, Paragraph 1)**

4. **Regarding Claim 2, 12, and 22,** Wilson discloses wherein each of said at least one data items is an operand. **(Wilson: Page 5, Section 1.3) The paper makes the simplification that objects being collected are from the variety of types possible and that it is easy to determine the type of an object.**

5. **Regarding Claim 3, 13, and 23,** Wilson discloses wherein said identifying step comprises: identifying a possible execution path leading to said execution point, wherein said possible execution path may be different from said actual execution path; performing a simulated execution of said possible execution path; and wherein said at least one data item roots and said at least one data items accessible to said processing task are identified by following said possible execution path to said current execution point. **Traversing the graph of pointer relationships, usually by either depth-first or breadth-first traversal. (Wilson: Page 9, Section 2.2, Paragraph 1)**

6. **Regarding Claim 4, 14, and 24,** Wilson discloses wherein said memory management operation comprises marking all of said memory areas that are accessible

to said processing task either directly or indirectly through said identified data items **(The objects that are reached are marked in some way; Wilson: Page 9, Section 2.2, Paragraph 1.)** and collecting unmarked memory areas for re-allocation during subsequent execution of said processing task. **(memory is swept ... find all unmarked objects and reclaim their space; Wilson: Page 9, Section 2.2, Paragraph 2)**

7. **Regarding Claim 5, 15, and 25,** Wilson discloses wherein said memory management operation comprises compacting said unmarked memory areas prior to reallocation. **(Wilson: Page 10, Section 2.3, Lines 1-7)**

8. **Claims 6, 16, and 26 are rejected using the same rationale as claim 3.**

9. **Regarding Claim 9, 19, and 29,** Wilson discloses wherein said processing task is a component of a computer program written in an object-oriented programming language. **(Wilson: Page 2, Section 1, Lines 23-28)**

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 10, 20, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilson ("Uniprocessor Garbage Collection Techniques").

12. **Regarding Claim 10, 20, and 30,** Wilson fails to disclose wherein said object oriented programming language is Java.

13. However, the Office takes Official Notice that it would have been obvious to a person of ordinary skill in the art to use Java as the object oriented programming language of Wilson.

14. The motivation for doing so would have been that Java allows the same program to be run on multiple operating systems, it contains built in support for networks, and it is a well understood and commonly accepted programming language among programmers.

15. Therefore, it would have been obvious to use Java as the object oriented programming language in Wilson for the benefits of allowing the program to run on multiple operating systems, containing built in support for networks, and being well understood and commonly accepted among programmers to obtain the invention as specified in claims 10, 20, and 30.

16. Claims 7, 8, 17, 18, 27, 28, and 31-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilson () as applied to claim 6 above, and further in view of Hosoya et al. ("Garbage Collection via Dynamic Type Inference" (herein after Hosoya)).

17. **Regarding Claims 7, 17, 27, and 31-33,** Wilson discloses scanning a plurality of program instructions corresponding to said programming task and logging a data type for each store instruction corresponding to each of said at least one data items; and simulating all possible execution paths up to said execution point for each of said at

least one data item root or said at least one data items. **Traversing the graph of pointer relationships, usually by either depth-first or breadth-first traversal. (Wilson: Page 9, Section 2.2, Paragraph 1)**

Wilson fails to disclose categorizing at least one of said data item roots or said at least one data items as a multiple-type variable if different data types are logged for different store instructions for a respective data item; determining the data type associated with each multiple-type variable at each of said plurality of program instructions for each of said possible execution paths; and checking said determined data type for each of said multiple-type variables at one of said plurality of program instructions corresponding to said current execution point; and said memory management operation is performed in dependence upon a result of said step of checking said determined data type.

Hosoya discloses categorizing at least one of said one or more data items as a multiple-type variable if different data types are logged for different store instructions for a respective data item; determining the data type associated with each multiple-type variable at each of said plurality of program instructions for each of said possible execution paths; and checking said determined data type for each of said multiple-type variables at one of said plurality of program instructions corresponding to said current execution point; and said memory management operation is performed in dependence upon a result of said step of checking said determined data type. **(Hosoya: Abstract, Lines 2-6 and Page 216, Lines 3-11)**

Wilson and Hosoya are analogous art because they are from the same field of endeavor, garbage collection.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to combine the type inference garbage collection of Hosoya into the garbage collection techniques of Wilson.

The motivation for doing so would have been that the garbage collection technique of Hosoya collects more garbage than any other algorithm using the same type system does. **(Hosoya: Page 233, Section 9, Lines 4-7)**

Therefore, it would have been obvious to combine the type inference garbage collection of Hosoya into the garbage collection techniques of Wilson for the benefit of collecting more garbage than any other algorithm using the same type system does to obtain the invention as specified in claims 17, 18, and 19.

18. **Regarding Claims 8, 18, and 28,** Wilson fails to disclose wherein said memory management operation involves tagging said at least one data item as suitable for reallocation if said determined data type is different for different ones of said possible execution paths at said current execution point.

Hosoya discloses wherein said memory management operation involves tagging said data item as suitable for reallocation if said determined data type is different for different ones of said possible execution paths at said current execution point. **(Hosoya: Abstract, Lines 2-6 and Section 1, Lines 1-5) Where Hosoya does not specifically say that semantic garbage will be reallocated it is understood that when speaking about garbage collection, garbage will be collected and reallocated.**

(10) Response to Argument

A. Regarding the lack of the suspension of an execution path, the applicant contends that Wilson teaches a mark-sweep technique that the Appellant discloses in the Appellant's specification. However, the only reference to mark and sweep techniques that occurs in the prior art section of the Appellant's specification merely states that the mark and sweep technique is a known algorithm for garbage collection. Also, just because the technique is mentioned in the Appellant's specification does not mean that the reference fails to teach the limitations disclosed in the Appellant's claims. This leads into the next point of contention.

The Appellant also contends that the Wilson reference discloses performing the mark-sweep technique prior to execution and therefore could not suspend an execution path. However, the mark-sweep technique as referenced in the Appellant's disclosure and as referenced in Wilson, is taught as an algorithm. (Wilson: Page 5, Section 1.4, Paragraphs 1-2.) Also, Wilson goes on, in that section, to state that incremental and generational schemes will be taught in sections 3 and 4, respectively. (Wilson: Page 5, Section 1.4, Paragraphs 3-4.) The significance of the statement that the mark-sweep technique is an algorithm means that it is a concept for implementing a larger garbage collection scheme. Therefore, the mark-sweep technique is not performed as the Appellant has alleged, and indeed is performed during the execution of the program.

Finally, the Appellant contends that the Examiner has improperly based an anticipation rejection on a combination of Wilson and an allegation of "known to one of skill in the art". However, the examiner has inartfully expressed the assertion, that

garbage collection occurs at a point in an execution path where execution is suspended in order to perform garbage collection, is inherent in the reference. In order to support the assertion of inherency the Examiner has provided the reference Gupta et al., herein after Gupta (attached herein). In Figures 1, 2, and 3 on pages 4 and 5 Gupta shows execution paths and the "Stop-the-world pause" caused by garbage collection schemes. More importantly, Gupta shows that even concurrent mark-sweep collectors contain a pause in execution. Gupta describes, on Page 3, in the section titled "Concurrent mark-sweep (CMS) collector", that the concurrent collector is "mostly concurrent", and shows in the figures that even a collector which is entitled concurrent still has a pause, or suspension, in the execution path to perform garbage collection. Gupta shows that all of the collectors possess this pause.

B. Regarding the lack of an identification step, the Appellant contends that because Wilson operates prior to execution and fails to suspend execution that it cannot disclose an identification step. However, as has been shown above, Wilson operates during execution and does perform a suspension step, therefore Wilson does teach an identification step as previously stated in the rejection at Wilson: Page 9, Section 2.2, Paragraph 1, specifically referring to the graphing of pointer relationships.

C. Regarding the lack of a determining step, the Appellant contends that because Wilson fails to teach the suspension of execution and the identification step that Wilson cannot teach a determining step. However, as shown above, Wilson does suspend execution and teaches a determining step, therefore Wilson teaches a determining step

as previously stated in the rejection at Wilson: Page 9, Section 2.2, Paragraph 1, specifically referring to the graphing of pointer relationships.

D. Regarding the lack of a memory management step, the Appellant contends that because Wilson fails to disclose the suspension step, the identification step, and the determining step that Wilson cannot disclose a memory management step. However, as shown above, Wilson does teach a suspension step, an identification step, and a determining step, therefore Wilson teaches a memory management step as previously stated in the rejection at Wilson: Page 9, Section 2.2, Paragraph 1.

E. Regarding the Wilson reference, the Appellant contends that the reference would lead one of ordinary skill in the art away from the Appellant's claimed invention. Appellant contends the Wilson teaches performing the mark-sweep technique of garbage collection prior to execution and therefore fails to disclose suspending the execution. However, the Appellant has failed to show any citations in Wilson that affirm this statement. Furthermore, the mark-sweep technique as referenced in the Appellant's disclosure and as referenced in Wilson, is taught as an algorithm. (Wilson: Page 5, Section 1.4, Paragraphs 1-2.) Also, Wilson goes on, in that section, to state that incremental and generational schemes will be taught in sections 3 and 4, respectively. (Wilson: Page 5, Section 1.4, Paragraphs 3-4.) The significance of the statement that the mark-sweep technique is an algorithm means that it is a concept for implementing a larger garbage collection scheme. Also, In Figures 1, 2, and 3 on pages 4 and 5 of Gupta, the reference shows execution paths and the "Stop-the-world pause" caused by garbage collection schemes. More importantly, Gupta shows that even concurrent

mark-sweep collectors contain a pause in execution. Gupta describes, on Page 3, in the section titled "Concurrent mark-sweep (CMS) collector", that the concurrent collector is "mostly concurrent", and shows in the figures that even a collector which is entitled concurrent still has a pause, or suspension, in the execution path to perform garbage collection. Gupta shows that all of the collectors possess this pause. Therefore, the mark-sweep technique of Wilson is not performed as the Appellant has alleged, and indeed is performed during the execution of the program. Therefore Wilson would not lead one of ordinary skill in the art away from the Appellant's claimed invention.

F. Regarding the rejection of claims 1-6, 9, 11-16, 19, 21-26, and 29 under 35 USC § 102, the Appellant contends there is no basis for rejection of the claims under 35 USC § 102 due to the arguments presented in Sections A-D. However, above the Examiner has traversed these arguments in their respective sections. Therefore, the rejections under 35 USC § 102 upheld.

G. Regarding the rejection of claims 10, 20, and 30, under 35 USC § 103, the Appellant contends that the rejections fail using the arguments presented in sections A-F. However, above the Examiner has traversed these arguments in their respective sections. Therefore, the rejections under 35 USC § 103 upheld.

H. Regarding the rejection of claims 7, 8, 17, 18, 27, 28, and 31-33 under 35 USC § 103, the Appellant contends that the rejections fail using the arguments presented in sections A-G. However, above the Examiner has traversed these arguments in their respective sections. Therefore, the rejections under 35 USC § 103 upheld.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Michael McFadden
Patent Examiner
Technology Center 2100

Conferees:

Lynne Browne
Appeal Practice Specialist, TQAS
Technology Center 2100

Hyung Sough
Supervisory Patent Examiner
Technology Center 2100

Attached:

Gupta et al., "Turbo-charging Java HotSpot Virtual Machine, v1.4.x to Improve the Performance and Scalability of Application Servers", http://java.sun.com/developer/technicalArticles/Programming/turbo/, Sun Microsystems, Inc., (archive.org date of Dec 08, 2003), pp.1-17.